**MODUL 4**

**DDL DAN DML**

# TUJUAN

* + Memahami SQL, komponen-komponen dan kegunaanya.
  + Mengenal operasi-operasi DDL dan DML.
  + Mampu melakukan operasi-operasi *retrieve* yang spesifik.
  + Mampu menguraikan dan mentranslasikan pernyataan-pernyataan SQL.

# PETUNJUK

* Awali setiap aktivitas dengan do’a semoga berkah dan mendapat kemudahan.
* Pahami tujuan, dasar teori dan latihan-latihan praktikum dengan baik dan benar.
* Kerjakan tugas-tugas praktikum dengan baik, sabar dan jujur.
* Tanyakan kepada asisten atau dosen apabila ada hal-hal yang kurang jelas.

1. **DASAR TEORI**

SQL terdiri dari beberapa komponen, dua di antaranya yang sangat fundamental adalah DDL (*Data Definition Language*) dan DML (*Data Manipulation Language*).

# DDL (Data Definition Language)

DDL menyediakan perintah-perintah yang berhubungan dengan operasi-operasi dasar seperti mendefinisikan, memodifikasi dan menghapus skema relasi— misalnya tabel, view, prosedur dan fungsi. Contoh perintah DDL adalah **CREATE, ALTER dan DROP.**

# DML (Data Manipulation Language)

DML menyediakan perintah-perintah untuk menambah, memodifikasi, menghapus dan mengambil data di dalam basis data. Contoh perintah DML adalah: **INSERT**, **UPDATE**, **DELETE** dan **SELECT**.

# Penambahan Data

Sintaks operasi penambahan data dapat dituliskan sebagai berikut:

INSERT INTO r (A1, A2, …, An) VALUES (V1, V2, …, An)

di mana:

r adalah relasi (tabel)

A adalah atribut (field atau kolom) V adalah value (nilai yang diisikan)

Misalkan tabel jurusan memiliki field id dan nama, maka pernyataan SQL untuk penambahan data dapat dituliskan sebagai berikut:

INSERT INTO jurusan (id, nama) VALUES (1, “Teknik Elektro”)

# Modifikasi Data

Sintaks operasi modifikasi data dapat dituliskan sebagai berikut:

UPDATE r

SET A1 = V1, A2=V2, …, An=Vn WHERE P

di mana:

*P* adalah predikat

Misalkan nilai field nama untuk dengan data id 1 ingin diubah menjadi Teknik Informatika, berikut pernyataan SQL-nya:

UPDATE jurusan

SET nama = “Teknik Informatika” WHERE id = 1



Perhatikan, operasi modifikasi tanpa menyertakan klausa **WHERE** akan mengakibatkan diubahnya seluruh data di tabel. Jadi, jika Anda hanya ingin memodifikasi data tertentu, gunakan selalu klausa **WHERE**.

# Penghapusan Data

Sintaks operasi penghapusan data dapat dituliskan sebagai berikut:

DELETE FROM r WHERE P

Misalkan nilai data dengan id ingin dihapus, pernyataan SQL-nya dituliskan sebagai berikut:

DELETE FROM jurusan WHERE id = 1



Perhatikan juga, operasi penghapusan tanpa menyertakan klausa **WHERE** akan mengakibatkan dihapusnya seluruh data di tabel. Jadi, jika Anda hanya ingin menghapus data tertentu gunkan selalu klausa **WHERE**.

# Seleksi Data

Sintaks operasi seleksi atau pengambilan data dapat dituliskan sebagai berikut:

SELECT A1, A2, …, An

FROM r1, r2, r3, …, rm WHERE P

Misalkan ingin menampilkan data id dan nama jurusan, pernyataan SQL-nya dituliskan sebagai berikut:

SELECT id, nama FROM jurusan

Sebagai alternatif, tersedia karakter asterisk (\*) yang menyatakan seluruh field. Jadi, pernyataan di bawah ini ekuivalen dengan pernyataan di atas

SELECT \*

FROM jurusan



Untuk efisiensi, karakter asterik sebaiknya digunakan jika ingin me-retrieve atau menampilkan seluruh nilai pada field.

# LATIHAN

1. **Menciptakan Tabel**

Membuat tabel jurusan dengan struktur sebagai berikut:

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| id | Number | Primary Key |
| nama | Text | Nama Jurusan |

Langkah – langkah:

* 1. Tuliskan pernyataan berikut ini pada jendela command prompt

CREATE TABLE jurusan ( id INT(3),

nama VARCHAR(30), PRIMARY KEY(id)

);

* 1. Periksa hasil dari pernyataan di atas dengan mengetikkan pernyataan berikut ini

SHOW tables;

Pada contoh di atas sengaja digunakan tipe data INT dan VARCHAR karena lebih general, Sebenarnya penggunaan tipe data NUMBER dan TEXT juga diterima, namun kurang standar.

# Menghapus Tabel

Untuk menghapus tabel, ikuti langkah-langkah berikut ini:

* 1. Tuliskan pernyataan berikut ini pada jendela command prompt

DROP TABLE jurusan;

* 1. Periksa hasil dari pernyataan di atas.

# Penambahan Data

Untuk menambahkan data pada suatu tabel, ikuti langkah-langkah berikut ini:

* 1. Tuliskan pernyataan berikut

INSERT INTO jurusan (id, nama) VALUES (1, “Teknik Elektro”)

* 1. Periksa hasil dengan mengetikkan perintah untuk menampilkan seluruh data yang berada pada tabel

SELECT \* FROM jurusan

# Modifikasi Data

Untuk memodifikasi data pada suatu tabel, ikuti langkah-langkah berikut ini:

* 1. Tuliskan pernyataan berikut

UPDATE jurusan

SET nama = "Teknik Informatika" WHERE id = 1;

* 1. Periksa hasil

# Penghapusan Data

Untuk menghapus data pada suatu tabel, ikuti langkah-langkah berikut ini:

* 1. Tuliskan pernyataan berikut

DELETE FROM jurusan WHERE id = 1;

* 1. Periksa hasil

# Seleksi Data

Untuk menyeleksi data pada suatu tabel, ikuti langkah-langkah berikut ini:

* 1. Tuliskan pernyataan berikut ini

SELECT id, nama FROM jurusan;

* 1. Amati hasil
  2. Bandingkan dengan pernyataan berikut ini

SELECT nama FROM jurusan;

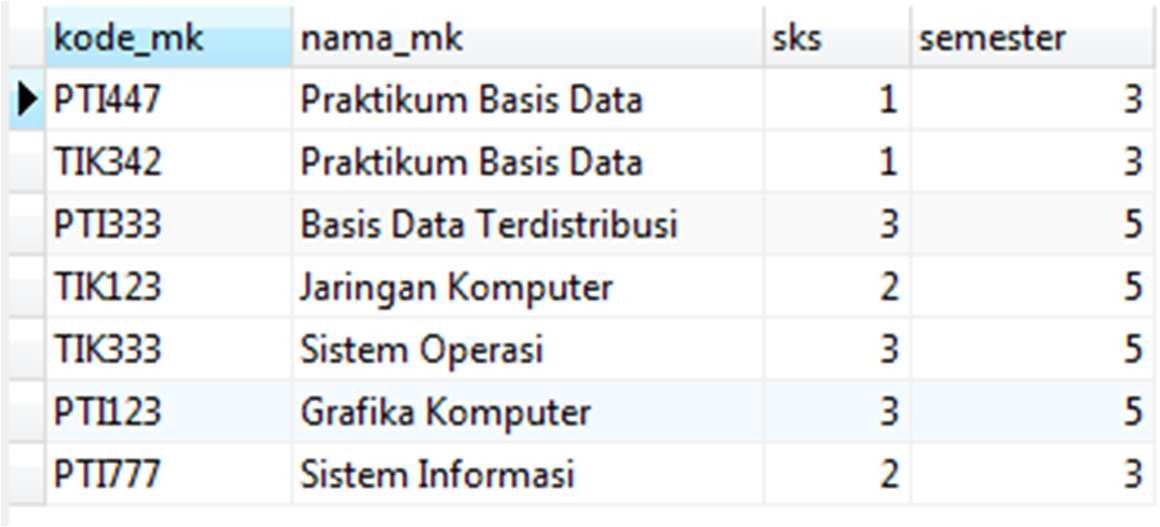
# Seleksi dengan Kriteria

Buat tabel **matakuliah** dengan menuliskan pernyataan berikut ini

CREATE TABLE matakuliah( kode\_mk char(12) not null, nama\_mk varchar(60) not null, sks int(1) not null, semester int(1) not null, primary key(kode\_mk)

);

Selanjutnya, isikan data sebagai berikut:



* 1. Dapatkan kode\_mk, nama\_mk, sks dan semester yang sksnya 3

SELECT \* FROM matakuliah WHERE sks = 3;

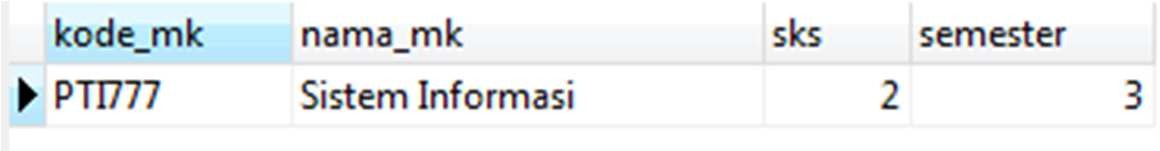
Hasil:



* 1. Dapatkan kode\_mk, nama\_mk, sks dan semester yang sksnya 2 dan semester 3

SELECT \* FROM matakuliah WHERE sks=2 AND semester=3;

Hasil:



* 1. Dapatkan kode\_mk, nama\_mk, sks dan semester yang sksnya kurang dari sama dengan 2.

SELECT \* FROM matakuliah WHERE sks <= 2;

Hasil:



# Seleksi dengan Operator Like

Operator LIKE memungkinkan pengambilan data yang mirip atau mendekati kata kunci.

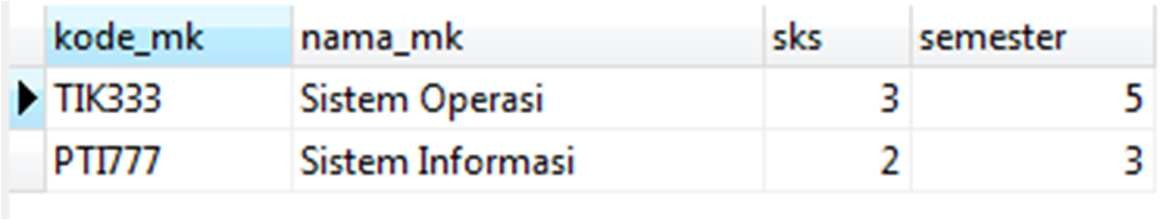
* 1. Dapatkan kode\_mk, nama\_mk, sks dan sem yang nama matakuliahnya berawalan “Sistem”.

SELECT \*

FROM matakuliah

WHERE nama\_mk LIKE “Sistem\*”;

Hasil:

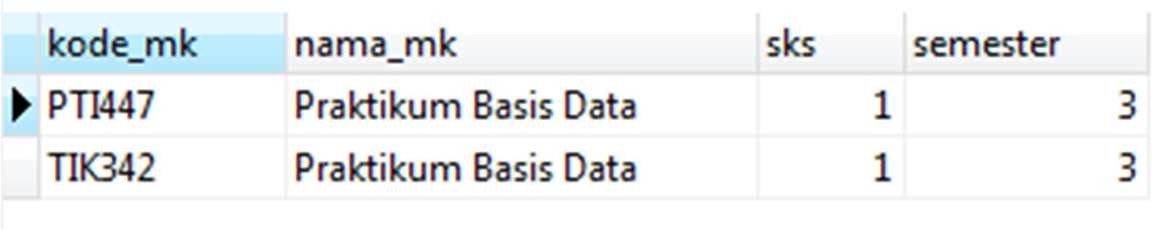


* 1. Dapatkan kode\_mk, nama\_mk, sks dan sem yang nama matakuliahnya berakhiran “Data”.

SELECT \*

FROM matakuliah

WHERE nama\_mk LIKE “\*Data”;

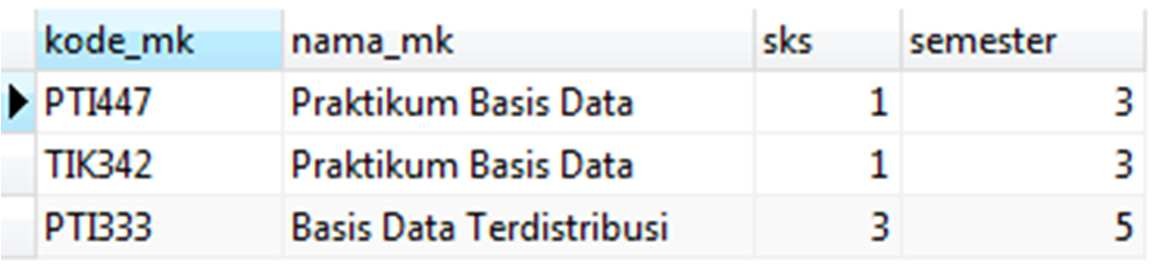
Hasil:

* 1. Dapatkan kode\_mk, nama\_mk, sks dan sem yang nama matakuliahnya mengandung kata “Basis”.

SELECT \*

FROM matakuliah

WHERE nama\_mk LIKE “\*Basis\*”;

Hasil:



Karakter wildcard di beberapa DBMS mungkin tidak sama. Sebagai contoh, MySQL atau PostgreSQL menggunakan tanda persen (%).

# Pengurutan Data

Untuk mengurutkan data, kita memanfaatkan klausa ORDER BY di mana nilai defaultnya adalah ASC (ascending atau pengurutan menaik). Contoh pengurutan data pada basis data adalah sebagai berikut:

* 1. Urutkan data matakuliah berdasarkan nama matakuliah dengan urutan menaik (ascending)

SELECT \*

FROM matakuliah ORDER BY nama\_mk;

* 1. Urutkan data matakuliah berdasarkan semester dengan urutan menurun (descending)

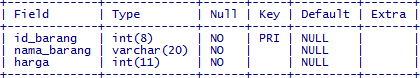
SELECT \*

FROM matakuliah ORDER BY sks DESC;

# TUGAS PRAKTIKUM

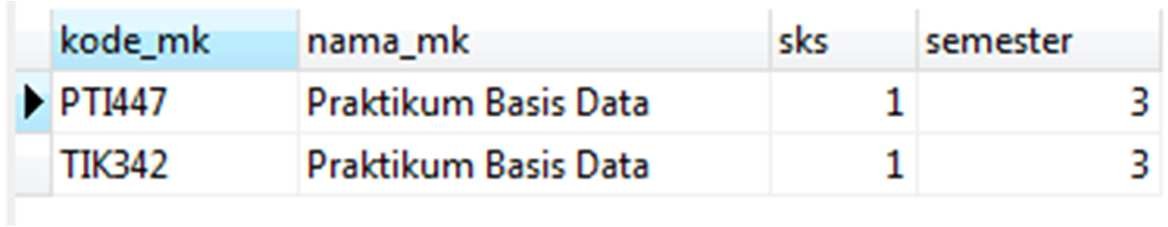
1. Tuliskan pernyataan SQL untuk menciptakan tabel dengan nama tabel **barang**

dengan deskripsi di bawah ini!



1. Tuliskan pernyataan SQL untuk mendapatkan data matakuliah dari tabel

**matakuliah** yang jumlah sksnya 4 atau 1! Hasil:



1. Tuliskan pernyataan SQL untuk mendapatkan data matakuliah dari tabel **matakuliah** yang nama depannya tidak sama dengan “Praktikum” dan sksnya 3! Hasil:



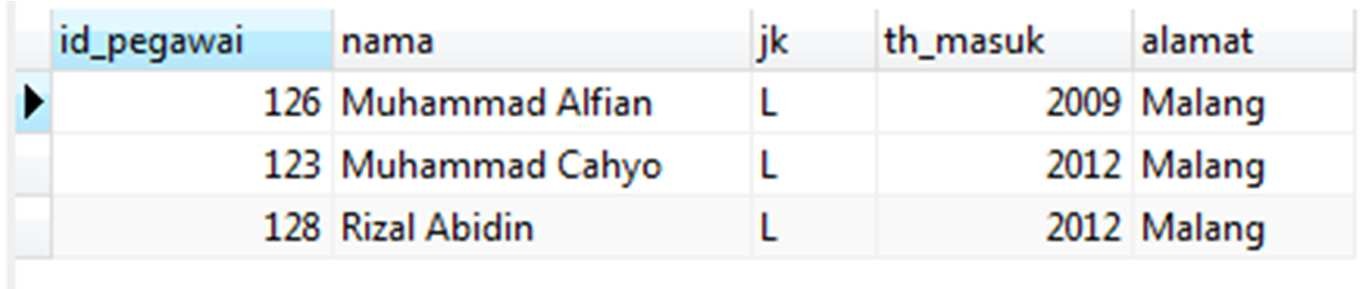
# TUGAS RUMAH

1. Tuliskan pernyataan SQL untuk membuat tabel **daftar\_pegawai** dengan isi tabel seperti di bawah ini:



**Note:** Tentukanlah tipe data yang sesuai dengan nama field yang ada, beserta dengan ukuran fieldnya!

1. Tuliskan pernyataan SQL untuk mendapatkan data pegawai yang tinggal di Malang dan berjenis kelamin laki-laki!

Hasil:

1. Tuliskan pernyataan SQL untuk mendapatkan data pegawai yang berjenis kelamin laki-laki yang masuk pada tahun diatas 2010 dan urutkan berdasarkan tahun terbesar! Hasil:



1. Tuliskanlah pernyataan SQL untuk mendapatkan data pegawai yang masuk sebelum tahun 2012 dan tinggal di Malang!

Hasil:



1. Berikan capaian pemahaman anda dalam bentuk persentase (0%-100%) tentang praktikum pertemuan ini!